STA 32 R Handout 5, Special Distributions

1 Bernoulli and Binomial Distribution

In R, there are three functions we will use that are associated with the Bernoulli or Binomial distribution:

- `dbinom(x,size,prob)`: Takes in the number of successes (`x`), the total sample size (`size`) and the probability of a success (`prob`). Returns the value

$$P(X = x) = \binom{size}{x} prob^x (1 - prob)^{size-x}$$

  `x` can also be a vector, in which case it returns the value of the p.d.f. for all given values of `x`.

- `pbinom(q,size,prob)`: Finds $P(X <= q)$, or the cumulative distribution function at the value `q`.

- `rbinom(n,size,prob)`: Generates `n` Binomial trials of size `size`, and returns the number of successes for each of the `n` trials.

Example: Suppose we are interested in the number of times we have to stop at a stoplight over a week. Assume the days are independent - that is assume that if you were stopped on one day, it has no effect on if you are stopped on any of the other days. The probability of getting stopped at this light is 0.70.
This is a Binomial random variable, where X is the number of times we are stopped at the stop light, where $X = 0, 1, \ldots 7$ for the 7 days.
To find the probability that $X = 3$:
`dbinom(3,7,0.70)`
To find the probability that $X \leq 3$:
`pbinom(3,7,0.70)`
To find the probability that $3 \leq X \leq 6$:
`sum(dbinom(c(3,4,5,6),7,0.70))`
To generate 100 random variables from this distribution (as if we had conducted this experiment over 100 weeks):

```
X = rbinom(100,7, 0.70)
```

2 Poisson Distribution

We will also use three functions for the Poisson Distribution

- `dpois(x,lambda)`: Takes in the number of times an event occurred (`x`), and the typical rate of occurrence (`lambda`). Returns the value

$$P(X = x) = e^{-\lambda}\frac{\lambda^x}{x!}.$$

  `x` can also be a vector, in which case it returns the value of the p.d.f. for all given values of `x`.

- `ppois(q,lambda)`: Finds $P(X \leq q)$, or the cumulative distribution function at the value `q`.

- `rpois(n,lambda)`: Generates `n` Poisson trials of with rate of occurrence `lambda`, and returns the number of times the event occurred for each of the `n` trials.

Example: The average number of homes sold by Acme Realty company is 2 homes per day. This is a Poisson distribution, with the typical rate of occurrence being 2. Let X be the number of houses sold.
To find the probability that $X = 3$ (exactly three houses are sold on any given day):
`dpois(3,2)`
To find the probability that $X \leq 3$ (at most three homes are sold on any given day):
`ppois(3,2)`
To find the probability that 0 or 1 homes are sold on a given day ($P(X = 0) + P(X = 1)$):
`sum(dpois(x(2,3),2))`
To generate 100 random variables from this distribution (as if we had observed how many homes were sold over 100 days):
`X = rpois(100,2)`

3 Bootstrapping

We have learned about the mean of the sample mean, $\mu_{\bar{X}}$, and the standard deviation of the mean, $s_{\bar{X}}$. We use the standard deviation of the sample mean because it is always smaller than the sample standard deviation. Sometimes, we would like an estimate of the standard deviation of the mean that is even smaller than $s_{\bar{X}}$, and we also want to improve our estimate of the sample mean. For this, we use bootstrapping.
The idea behind bootstrapping is to take our sample of data (of size n), and with it generate similar "new" samples by taking n observations with replacement from our original sample. With these "new' samples, we may estimate the mean and variance of the sample mean more accurately.
There are two types of bootstrap - parametric and non-parametric.

3.1 Non-parametric

The main steps for **non parametric** bootstrapping will be as follows:

1. From the original data $X_1, \ldots, X_n$ (our n data points), re-sample n new data points **with replacement** from $X_1, \ldots, X_n$, and call the new n data points one bootstrap sample, with notation $X^*_{1i}, \ldots, X^*_{1n}$.
   Example: If our original data points were 23, 30, 35, 24, 32, our bootstrap sample could be 23, 24, 30, 30, 23.
   Create B total bootstrap samples of size n. Usually B is at least 100.

2. For each of the bootstrapped samples, calculate the mean. We will end up with B total mean values, $\bar{X}^*_1, \ldots \bar{X}^*_B$.

3. To calculate the bootstrap standard deviation, calculate the standard deviation of the B mean values. This is in general, smaller than $s_{\bar{X}}$.

4. To calculate the bias of $\bar{X}$, calculate the average of all B bootstrap means $\bar{X}^*$, then $bias = \bar{X}^* - \bar{X}$ where is the sample mean of our original data.

3.2 Parametric

The main steps for **parametric** bootstrapping are nearly identical - only the sampling step changes. Here we assume we know the true underlying distribution - for example we assume we sampled from a binomial distribution.

1. From the original data $X_1, \ldots X_n$, estimate the parameters of the underlying distribution. For example, if we have a sample of 0's and 1's, we would estimate the proportion of successes as (sum of all the 1's)/n.
   Randomly generate n data points from the underlying distribution, using the estimates found above. This is one bootstrap sample.
   Example: If our original data points were 0, 1, 0, 1, 0, 0, we would randomly generate from a $Binom(2/6, 6)$ distribution using `rbinom`.
   Create B total bootstrap samples of size n. Steps 2-4 are exactly the same.

Notice that non-parametric bootstrapping does not require that we know the underlying distribution. However, if we can confidently state the underlying distribution, parametric bootstrapping will typically give smaller standard errors.

3.3 Example

Lets assume we have the following data:

```
X = c(8, 5, 4, 5, 3, 5, 6, 3, 1, 0, 3, 3, 5, 1, 4, 4, 2, 0, 4, 6)
```

We can calculate the sample mean, and the standard deviation of the sample mean easily.

```
n = length(X)
Xbar = mean(X)
sXbar = sd(X)/sqrt(n)
```

Now, lets calculate the mean of 100 bootstrap samples:

```
B= 100
BootStrapMean = sapply(1:B,function(i){
   NewSample = sample(X,n,replace = TRUE)
   MeanOfNew = mean(NewSample)
   return(MeanOfNew)
   })
```

Lets see what the bootstrap standard deviation of the mean is, and the mean of the bootstrap means:

```
sd(BootStrapMean)
mean(BootStrapMean)
```

We should see that in some cases, the bootstrap standard deviation is already lower than the standard deviation of the mean. If it is not, we can increase the bootstrap sample size (increase B) in order to make our estimate more accurate.

3.4 When does bootstrapping fail?

Bootstrapping is a good idea if you do not have standard theory to back up your results, or if you are estimating something very complicate (i.e., something other than the mean). It is a bad idea if:

1. You believe your sample is non-representative.
2. You have a small sample size (usually less than 50).

These conditions basically both say the same thing: we are not confident in our sample, so using information by resampling would not yield much gain.